AMENDMENTS TO THE CLAIMS

Applicants submit below a complete listing of the current claims, including marked-up claims with insertions indicated by underlining and deletions indicated by strikeouts and/or double bracketing. This listing of claims replaces all prior versions and listings of claims in the application:

1. (Currently amended) A computer-implemented method of generating a componentized user interface for one or more computer applications, at least one of which is capable of performing one or more functions, the method comprising:

displaying a first set of interface elements provided by a framework;

displaying a second set of interface elements provided by a first plug-in that is linked to the framework;

displaying a third set of interface elements provided by a second plug-in that is linked to the framework;

providing an interface between the one or more computer applications and the first ~~plug-in~~ and second plug-ins ~~and between the one or more applications and the second plug-in with a shell adapter,~~ in order to utilize the second set of interface elements and the third set of interface elements, ~~wherein the shell adapter~~ the interface comprising:

a first application specific adapter that maps interface elements of the first plug-in ~~and interface elements of the second plug-in~~ to functions of the one or more computer applications;

a second application specific adapter that maps interface elements of the second plug-in to functions of the one or more computer applications; and

a shell adapter that interfaces between the framework and the first and second application specific adapters; and

in response to a user activating an interface element provided by the first plug-in ~~or the second plug-in~~, causing a computer application of the one or more computer applications to

perform a function, wherein the [[shell]] first application specific adapter maps the interface element to the function.

2. (Previously presented) The computer-implemented method of claim 1, wherein the first plug-in comprises:

(i) a first file that provides an interface between the framework and the first plug-in; and

(ii) a second file that is written in a markup language and that includes interface elements.

3. (Previously presented) The computer-implemented method of claim 2, wherein the interface elements are selected from a group comprising a toolbar, a status bar, and a menu bar.

4. (Previously presented) The computer-implemented method of claim 1, wherein the second plug-in comprises:

(i) a first file that provides an interface between the framework and the second plug-in; and

(ii) a second file that is written in a markup language and that includes interface elements.

5. (Previously presented) The computer-implemented method of claim 4, wherein the interface elements are selected from a group comprising a toolbar, a status bar, and a menu bar.

6. (Original) The computer-implemented method of claim 1, wherein the framework is configured to discover the first plug-in and the second plug-in.

7. (Original) The computer-implemented method of claim 6, wherein the framework further comprises a user interface component loader to load the first plug-in and the second plug-in.

8. (Previously presented) The computer-implemented method of claim 2, wherein the first file comprises an executable file and the second file comprises information written in an extensible markup language (XML).

9. (Previously presented) The computer-implemented method of claim 2, wherein the first file comprises an executable file and the second file comprises information written in a standard generalized markup language (SGML).

10. (Previously presented) The computer-implemented method of claim 4, wherein the first file comprises an executable file and the second file comprises information written in an extensible markup language (XML).

11. (Previously presented) The computer-implemented method of claim 4, wherein the first file comprises an executable file and the second file comprises information written in a standard generalized markup language (SGML).

12. (Canceled)

13. (Previously presented) The computer-implemented method of claim 1, wherein both the second set and the third set of interface elements comprise interface elements for a first application.

14. (Original) The computer-implemented method of claim 1, wherein the second set of interface elements comprises interface elements for a first application and the third set of interface elements comprise interface elements for a second application that is different from the first application.

15. (Currently amended) A computer implemented method of providing extensibility to a user interface for one or more computer applications, at least one of which is capable of performing one or more functions, the method comprising:

providing a framework, the framework comprising a first set of interface elements and a user interface component loader, the framework configured to discover a plug-in located in a plug-in directory;

loading the plug-in with the user interface component loader, the plug-in to provide a second set of interface elements;

providing an interface between the one or more computer applications and the plug-in ~~with a shell adapter~~ in order to utilize the second set of interface elements, ~~wherein the shell adapter~~ the interface comprising:

an application specific adapter that maps interface elements of the plug-in to functions of the one or more computer applications; and

a shell adapter that interfaces between the framework and the application specific adapter; and

in response to a user activating an interface element provided by the plug-in, causing a computer application of the one or more computer applications to perform a function, wherein the [[shell]] application specific adapter maps the interface element to the function.

16. (Previously presented) The computer-implemented method of claim 15, wherein the plug-in comprises:

(i) a first file that provides an interface between the framework and the plug-in; and

(ii) a second file that is written in a markup language and that includes interface elements.

17. (Previously presented) The computer-implemented method of claim 16, wherein the interface elements are selected from a group comprising a toolbar, a status bar, and a menu bar.

18. (Previously presented) The computer-implemented method of claim 16, wherein the first file comprises an executable file and the second file comprises information written in an extensible markup language (XML).

19. (Previously presented) The computer-implemented method of claim 16, wherein the first file comprises an executable file and the second file comprises information written in a standard generalized markup language (SGML).

20. (Original) The computer-implemented method of claim 15, wherein the framework is configured to provide the first set of interface elements for a plurality of applications.

21. (Currently amended) The computer-implemented method of claim 15, wherein the method further comprises:

loading a second plug-in with the user interface component loader, the second plug-in to provide a third set of interface elements;

providing an interface between the one or more applications and the second plug-in with a second [[shell]] application specific adapter in order to utilize the third set of interface elements.

22. (Previously presented) The computer-implemented method of claim 21, wherein both the second set and the third set of interface elements comprise interface elements for a first application.

23. (Original) The computer-implemented method of claim 21, wherein the second set of interface elements comprises interface elements for a first application and the third set of interface elements comprise interface elements for a second application that is different from the first application.

24. (Canceled)

25. (Canceled)

26. (Currently amended) A computer storage medium bearing instructions, which, when executed, carry out a method for generating a componentized user interface for a plurality of computer applications or hosting environments, at least one of which is capable of performing one or more functions, the method comprising:

providing a first set of interface elements with a user interface framework, the interface elements comprising at least one menu;

providing a second set of interface elements with a plug-in, the second set of interface elements comprising at least one of a toolbar, a status bar, and a menu bar;

providing a first interface between the plug-in and a first computer application or hosting environment, wherein the first interface maps interface elements of the plug-in to functions of the first computer application or hosting environment;

in response to a user activating a first interface element provided by the plug-in, causing the first computer application or hosting environment to perform a first function, wherein the first interface maps the first interface element to the first function;

providing a second interface between the plug-in and a second computer application [[of]] or hosting environment, wherein the second interface maps interface elements of the plug-in to functions of the second computer application or hosting environment; [[and]]

in response to the user activating a second interface element provided by the plug-in, causing the second computer application or hosting environment to perform a second function, wherein the second interface maps the second interface element to the second function; and

providing a shell adapter interface between the user interface framework and the first interface and the second interface, whereby the framework is configured to provide the first set of interface elements in conjunction with the first computer application or hosting environment and the second computer application or hosting environment.

27. (Canceled)